VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi – 590018

PROJECT REPORT

on

“CROP PRICE PREDICTION USING AI&ML”

Submitted in partial fulfilment for the award of the degree

Bachelor of Engineering

in

Computer Science and Engineering

Submitted by

SAURAV KUMAR	**1ST20CS103**
PRAFULL KUMAR JHA	**1ST20CS084**
PRASHANT S PATIL	**1ST20CS087**
SHREYAS P BIDARI	**1ST20CS114**

Under the Guidance of

Dr. Asha S R
Assistant Professor
Department of CSE

SAMBHRAM
INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SAMBHRAM INSTITUTE OF TECHNOLOGY

M. S. Palya, Bengaluru – 560097

2023-2024

SAMBHRAM INSTITUTE OF TECHNOLOGY

M. S. Palya, Bengaluru – 560097

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the Project work entitled "**Crop Price Prediction Using AI&ML**" carried out by Mr./Ms. **SAURAV KUMAR 1ST20CS103, PRAFULL KUMAR JHA 1ST20CS084, PRASHANT S PATIL 1ST20CS087, SHREYAS P BIDARI 1ST20CS114,** bonafide student of **SAMBHRAM INSTITUTE OF TECHNOLOGY** in partial fulfilment for the award of **BACHELOR OF ENGINEERING IN COMPUTER SCIENCE AND ENGINEERING** of **VISVESVARAYA TECHNOLOGICAL UNIVERSITY**, Belagavi during the year **2023-2024**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The Project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

Dr. Asha S R Assistant Professor Dept. of CSE SaIT, Bengaluru	**Dr. T. John Peter** HOD Dept. of CSE SaIT, Bengaluru	**Dr. H. G. Chandrakanth** Principal SaIT, Bengaluru

EXTERNAL VIVA:

Name of the Examiners | **Signature with date**

1.

2.

SAMBHRAM INSTITUTE OF TECHNOLOGY

M. S. Palya, Bengaluru – 560097

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We the students of 8th Semester, Dept. of CSE, SaIT doing the Final Year Project, declare that

[1] The Hardware/Software is not purchased/brought from any outside originations.

[2] The Hardware/Software is not from any other previous final year engineering projects of VTU.

[3] Our Project work is as per VTU norms, and we have followed the rules and regulations.

Violating any of the above conditions, we will accept the action taken by the Department/College/ VTU in this regard.

The Title of the Project is

"CROP PROCE PREDICTION USING AI&ML".

The project work is guided by:

Dr. ASHA S R

Asst. Professor

Dept. of CSE, SaIT

No.	Student Name	USN	Signature
1	Saurav Kumar	1ST20CS103	
2	Shreyas P Bidari	1ST20CS114	
3	Prafull Kumar Jha	1ST20CS084	
4	Prashant S Patil	1ST20CS087	

ABSTRACT

Farmers are key actors in the agricultural sector because they cultivate food and economic stability. On the other hand, one of their major problems is that crop prices fluctuate a lot in most cases after harvesting. These price fluctuations can have very serious consequences for farmers, leading to massive losses. Besides, these price changes also affect overall economic stability due to the huge contribution made by agriculture to the GDP of many nations. It therefore becomes essential to create enabling conditions for rural communities with regard to making informed choices about crop selection as well as the best timing for planting and harvesting. This could be done through assessing and approximating prices of crops. By being able to accurately predict different crop prices, farmers stand a chance of mitigating against the risks associated with changing market trends thus reducing their losses and improving on farming practices. Our study aims at developing an empirical model capable of predicting crop prices within this framework. The model includes diverse features based on historical rainfall data and Wholesale Price Index (WPI) data analysis especially. The Decision Tree algorithm is chosen because it can handle both numerical and categorical data, thereby making it ideal for studying the wide range of factors affecting crop prices. It is able to predict future price trends by learning from historical data about rainfall, WPI and crop prices. In addition to just forecasting for the next season's prices this model has a broader use as well. Our objective is to give farmers an idea of how prices may change in the course of the year so that they can organize their farming activities and marketing strategies accordingly. Thus, We empower our farmers with information on how to cope with changing market conditions and how they can improve their agricultural practices for higher efficiency and profitability levels. In general, through our research we attempt to fill in the gap between analytics and decision making in agriculture which enables farmers with actionable insights necessary in navigating crop price volatility complexities. We work towards having a resilient sustainable agricultural sector through predictive modeling hence contributing to food security and economic development at large

ACKNOWLEDGEMENT

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders and friends. A number of personalities, in their own capacities have helped us in carrying out this project work. We would like to take this opportunity to thank them all.

We would like to express our heartfelt thanks to **Dr. H. G. Chandrakanth, Principal, Sambhram Institute of Technology**, whose valuable guidance has been the one that helped us to complete the project.

We would like to express our profound gratitude to **Dr. T. John Peter, HOD, Department of CSE, Sambhram Institute of Technology**, for his suggestions and his instructions have served as the major contribution towards the completion of the project.

We would like to extend our impassioned thanks and admiration to our **guide, Dr. Asha S R, Assistant Professor, Department of CSE, Sambhram Institute of Technology**, for their able guidance, regular source of encouragement and assistance throughout this project.

We would like to thank all the teaching and non-teaching staff members of the Computer Science Department, who have helped us directly or indirectly for the successful completion of the project.

Finally, we would like to thank our Parents and Friends who have helped us with their valuable suggestions and guidance for the completion of our project.

SAURAV KUMAR	**1ST20CS103**
PRAFULL KUMAR JHA	**1ST20CS084**
PRASHANT S PATIL	**1ST20CS087**
SHREYAS P BIDARI	**1ST20CS114**

TABLE OF CONTENTS

Chapter	Title	Page No.
1	**INTRODUCTION** 1.1 Machine Learning 1.2 Types Of Machine Learning 1.3 Regression 1.4 Decision Tree Regressor Algorithm 1.5 Objectives 1.6 Challenges 1.7 Proposed System 1.8 System Process	**1-8**
2	**LITERATURE SURVEY**	**9-11**
3	**SYSTEM REQUIREMENTS** 3.1 Functional Requirements 3.2 Non-Functional Requirements	**12-14**
4	**SYSTEM DESIGN** 4.1 System Architecture 4.2 Flow Chart 4.3 Activity Diagram 4.4 Use Case Diagram 4.5 Sequence Diagram	**15-21**
5	**IMPLEMENTATION** 5.1 Frontend And Backend Development 5.2 Source Code	**22-39**
6	**RESULTS**	**40**
7	**SNAPSHOTS**	**41-50**
8	**CONCLUSION**	**51**
9	**SCOPE FOR FUTURE WORK**	**52**
10	**BIBILOGRAPHY**	**53**
11	**REFERENCES**	**54**

LIST OF FIGURES

Fig No.	Name	Page No.
1.1	Decision Tree	4
4.1	System Architecture	15
4.2	Flow Chart	16
4.3	Activity Diagram	18
4.4	Use Case Diagram	19
4.5	Sequence Diagram	20
4.6	UML Diagram	21
7.1	Sample Dataset Of Barley	41
7.2	Homepage of the Web Application	42
7.3	The List Of Top Gainers	42
7.4	The List Of Top Loosers	43
7.5	Brief Information On Maize Crop	43
7.6	Maize Crop's 12 months Prediction (with graph)	44
7.7	Brief Information On Rape Crop	45
7.8	Rape Crop's 12 months Prediction (with graph)	46
7.9	Brief Information On Wheat Crop	47
7.10	Wheat Crop's 12 months Prediction (with graph)	48
7.11	Brief Information On Copra Crop	49
7.12	Copra Crop's 12 months Prediction (with graph)	50

CHAPTER 1

INTRODUCTION

1.1 What is **Machine Learning ?**

Machine learning (ML) is the study of computer algorithms that improve automatically through experience. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop conventional algorithms to perform the needed tasks. Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a related field of study, focusing on exploratory data analysis through unsupervised learning. In its application across business problems, machine learning is also referred to as predictive analytics.

Machine Learning (ML) has proven to be one of the most game-changing technological advancements of the past decade. In the increasingly competitive corporate world, ML is enabling companies to fast-track digital transformation and move into an age of automation. Some might even argue that AI/ML is required to stay relevant in some verticals, such as digital payments and fraud detection in banking or product recommendations. The eventual adoption of machine learning algorithms and its pervasiveness in enterprises is also well-documented, with different companies adopting machine learning at scale across verticals. Today, every other app and software all over the Internet uses machine learning in some form or the other. Machine Learning has become so pervasive that it has now become the go-to way for companies to solve a bevy of problems.

1.2 Types Of Machine Learning

Based on the methods and way of learning, machine learning is divided into mainly four types, which are:

1. Supervised Machine Learning
2. Unsupervised Machine Learning
3. Semi-Supervised Machine Learning
4. Reinforcement Learning

1.2.1 Supervised Machine Learning

As its name suggests, Supervised machine learning is based on supervision. It means in the supervised learning technique, we train the machines using the "labelled" dataset, and based on the training, the machine predicts the output. Here, the labelled data specifies that some of the inputs are already mapped to the output. More preciously, we can say; first, we train the machine with the input and corresponding output, and then we ask the machine to predict the output using the test dataset.

Let's understand supervised learning with an example. Suppose we have an input dataset of cats and dog images. So, first, we will provide the training to the machine to understand the images, such as the shape & size of the tail of cat and dog, Shape of eyes, colour, height (dogs are taller, cats are smaller), etc. After completion of training, we input the picture of a cat and ask the machine to identify the object and predict the output. Now, the machine is well trained, so it will check all the features of the object, such as height, shape, colour, eyes, ears, tail, etc., and find that it's a cat. So, it will put it in the Cat category. This is the process of how the machine identifies the objects in Supervised Learning.

The main goal of the supervised learning technique is to map the input variable(x) with the output variable(y). Some real-world applications of supervised learning are Risk Assessment, Fraud Detection, Spam filtering, etc.

Categories of Supervised Machine Learning

Supervised machine learning can be classified into two types of problems, which are given below:

- Classification
- Regression

1.3 Regression

Regression algorithms are used to solve regression problems in which there is a linear relationship between input and output variables. These are used to predict continuous output variables, such as market trends, weather prediction, etc.

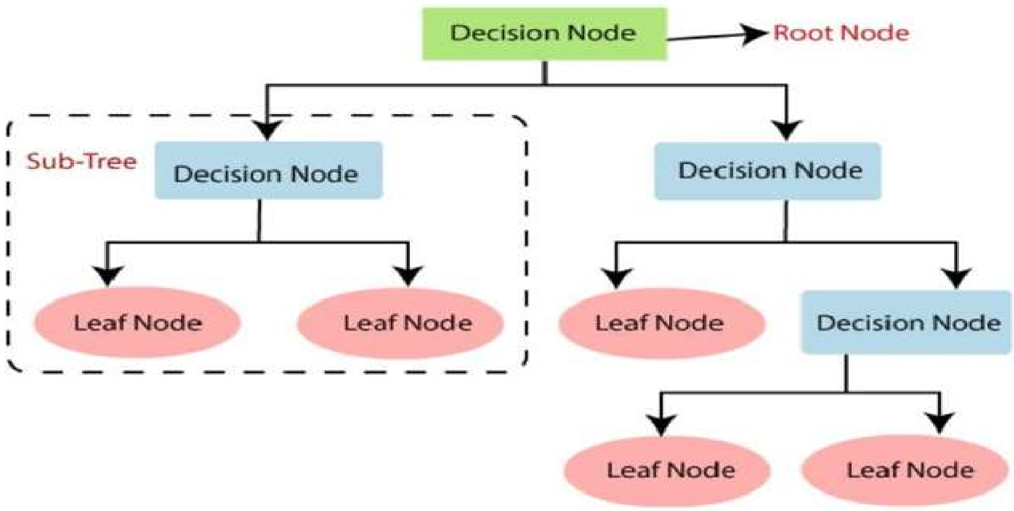
Some popular Regression algorithms are given below:

- Simple Linear Regression Algorithm
- Multivariate Regression Algorithm
- Decision Tree Algorithm
- Lasso Regression

1.4 Decision Tree Regressor Algorithm

- Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems.
- It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.
- In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node.
- Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
- The decisions or the test are performed on the basis of features of the given dataset.
- How does the Decision Tree algorithm Work?

- Step-1: Begin the tree with the root node, says S, which contains the complete dataset.
- Step-2: Find the best attribute in the dataset using Attribute Selection Measure (ASM).
- Step-3: Divide the S into subsets that contains possible values for the best attributes.
- Step-4: Generate the decision tree node, which contains the best attribute.
- Step-5: Recursively make new decision trees using the subsets of the dataset created in
- step -3. Continue this process until a stage is reached where you cannot further
- classify the nodes and called the final node as a leaf node.



DECISION TREE

Fig 1.1 DECISION TREE

1.4.1 **Advantages Of Decision Tree**

- Simple to understand and to interpret. Trees can be visualized.
- Requires little data preparation. Other techniques often require data normalization, dummy variables need to be created and blank values to be removed. Note however that this module does not support missing values.
- The cost of using the tree (i.e., predicting data) is logarithmic in the number of data points used to train the tree.
- Able to handle both numerical and categorical data. However, the scikit-learn implementation does not support categorical variables for now. Other techniques are usually specialized in analyzing datasets that have only one type of variable. See algorithms for more information.
- Able to handle multi-output problems.
- Uses a white box model. If a given situation is observable in a model, the explanation for the condition is easily explained by boolean logic. By contrast, in a black box model (e.g., in an artificial neural network), results may be more difficult to interpret.
- Possible to validate a model using statistical tests. That makes it possible to account for the reliability of the model.
- Performs well even if its assumptions are somewhat violated by the true model from which the data were generated.

1.5 Objectives

The central focus of the project is to predict crop price based on the previous trends in weather, yield and price. In agri-based country like India, predicting crop yield and price can aid a lot of people whose sole survival is dependent on the crop that they plan to grow.

The major objectives of the proposed system include :

- To collect historical crop prices and weather related data.
- To perform data pre-processing on the gathered data.
- To analyze the trends in the datasets.
- To build a predictive model using Decision Tree Regressor Algorithm.
- To develop a web application to display the predicted prices.

1.6 Challenges

1.6.1 Proneness to Errors

Evidence has shown that ML is autonomous but highly susceptible to both machine and human errors. For instance, if the right amount of data is not trained in the dataset, biased predictions can be made and generalized. This is because the quality of prediction would largely depend on the correctness of the training dataset. Prediction errors in ML often appear which are very difficult to diagnose and correct because they require rigorous underlying complexities of the algorithms and associated processes.

1.6.2. Data Acquisition

ML algorithms rely on data to learn and predict academic events. It is data hungry. However, the acquisition of these data is not easy, but the larger the data, the better the machine learning and prediction reliability.

1.6.3. Time Factors

Machine learning algorithms require sufficient time to train and learn to be able to function effectively. This could lead to time wastage. The ML forecasting technique, on the other hand, necessitates patience and time to assure accuracy. The time complexity could increase depending on the size of data. The more the data are, the more time would be required.

1.6.4. Verification Challenges

Sometimes, it is difficult to verify certain facts that are not included in the historical data, which means that ML predictions may not be accurate in certain cases. It could also take some time for the model to be trained or master the dataset for effective decision.

1.7 Proposed System

In our work, we will be using Decision Tree Algorithm. Decision Tree is a Supervised learning technique that can be used for both classification and Regression problem. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.Decision trees are easier to interpret than random forests since they are straightforward. It is easy to visualize a decision tree and understand how the algorithm reached its outcome.

1.8 System Process

The steps involved in building our model are:

1. Data Gathering Dataset is prepared by collecting the crop-related information, such as month, year, rainfall, whole price index(WPI) from data.gov.in.

2. Data Pre-Processing Data Pre-Processing involves preparing the raw data and making it suitable for a machine learning model. Data Cleaning was done to handle missing data and noisy data.

3. Data Exploration Data Exploration is visualizing the datasets in scatter plots, bar plots etc. This allows us to identify the trends in the data and obtain repeated trends ,spots anomalies etc. to decide the best way to monitor data sources to get the results with greater precision.

4. Model Building The process of modeling means training a Machine Learning Algorithm to predict the labels from the features, tuning it for our need, and validating it on the hold out data. The output from modeling is a trained model that can be used for inference, making predictions on new data points.

We can build our model using Decision Tree algorithm: Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems.

It is a treestructured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

5. Web Application The predicted prices are displayed on the web application developed using Python's flask framework. A home page is created from where we can explore 23 different crops and see their forecasts for the next 12 months.

CHAPTER 2

LITERATURE SURVEY

- Monali Paul, Santosh K Vishwakarma, Ashok Verma

  The purpose of this article is to anticipate crop yields, and crops are classified and examined. Data mining algorithms such as KNN and Naive Bayes are used to classify the data. The use of data mining to develop our idea will be advantageous.

- Abdullah Na, William Isaac, Ekaram Khan

  This study features a Smartphone app that will measure soil pH, temperature, and humidity in real time. Using a variety of methodologies, this study aids in remote soil analysis.

- S. Nagini, Dr.T.V. Rajnikanth, B.V. Kiranmayee

  This work proposes a method for conducting exploratory data analysis to develop multiple prediction models. To anticipate good crops, several regression approaches, such as linear regression, are applied. Various machine learning techniques are used to predict the best yield for the farmer.

- Awanit Kumar, Shiv Kumar

  The article suggested methods for predicting crop production for the coming year. The system employs fuzzy logic-based prediction methods. Fuzzy logic is a rule-based forecasting logic in which rules are applied to the land for agriculture, rainfall, and crop forecasting. K-means can be used to analyse the received data set by the system.

- Pooja More, Sachi Nene

  To anticipate the most suited crops, this paper combines modern artificial neural network technology and machine learning algorithms such as SVM and linear regression.

- Rakesh Kumar I, M.P. Singh

  CSM (crop selection method) and machine learning algorithms were proposed in the paper as strategies for appropriate crop selection. It is primarily concerned with boosting the farmer's profit by picking suitable crops.

- Ishita Ghutake, Ritesh Verma, Rohit Chaudhari ,and Vidhate Amarsinh

  The model uses crop and rainfall data from data.gov.in and applies decision Tree Regressor Algorithm to predict the price of the crops with 92% accuracy and it aims at a farmer friendly interactive website, predicting the prices through web application.

- Pandit Samuel, B.Saahithi, T.Saheli, D.Ramanika, N.Anil Kumar

  Different data mining techniques were used to build a precise model for price prediction.ML algorithms such as Linear Regression , decision Trees , XGBoost and Neutral Networks were used among which XGBoost gave good performance.

- S.Rajeshwari,K.Suthendran

  Using the previous year month-wise average commodity prices , HADT algorithm was applied to predict agricultural product prices with 90% accuracy.

- Gangasagr HL,Jovin Dsouza,Bhagyashree B Yargal,Arun Kumar

  The machine learning algorithm of Random Forest Regressor gave the best results with an accuracy of over 92% when compared to Linear Regression, Decision Tree Regressor and Support Vector Regressor(SVR).

- Aryamol A U,Raji P G,Sam G Benjami

  Considering various factors -PH,Nitrogen Levels and nutrient essentials in soil ,XG Boost was implemented to develop the model and it is achieved 89% accuracy in predictions.

- P.Priya,U.Muthaiah and M. Balamurgan

  Developed a predictive model using recorded data of Tamil Nadu's previous year crop yield to predict the yield of the crop in present. They applied Decision Tree algorithm and predicted the yield of the crop with 90% accuracy.

CHAPTER 3

SYSTEM REQUIREMENTS

System Requirements Specification may be a document or set of documentation that describes the features and behavior of a system or software application. It includes a range of elements that attempts to define the intended functionality required by the users to satisfy their different users.

3.1 FUNCTIONAL REQUIREMENTS

- The Functional Requirements Definition reports and tracks the fundamental information expected to effectively portray business and handy necessities.
- The Functional Requirements Definition report is created within the midst of the design Phase of the endeavor.
- Its objective gathering is that the endeavor boss, errand gathering, wander bolster, client/customer, and any accomplice whose information/respect into the necessities definitions system is required.

3.2 NON-FUNCTIONAL REQUIREMENTS

- Non-Functional Requirement (NFR) specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system.
- Failing to meet non-functional requirements can result in systems that fail to satisfy user needs. Nonfunctional Requirements allows you to impose constraints or restrictions on the design of the system across the various agile backlogs.
- Example, the site should load in 3 seconds when the number of simultaneous users are > 10000.
- They specify the criteria that can be used to judge the operation of a system rather than specific behaviours. They may relate to emergent system properties such as reliability, response time and store occupancy.

3.2.1. Hardware Requirements

Hardware specifications are technical descriptions of the computer's components and capabilities. Processor speed, model and manufacturer. Processor speed is usually indicated in gigahertz (GHz). the upper the quantity, the faster the pc. Random Access Memory (RAM). this is often typically indicated in gigabytes (GB). The more RAM in an exceedingly computer the more it can do simultaneously. fixed disk (sometimes called ROM) space. this is often typically indicated in gigabytes (GB).

- Processor Intel Pentium/Core – 1.7GHz and above
- Memory 1GB and above
- Storage 80GB minimum fee space
- Graphics 1GB and above 5.4

3.2.2. Software Requirements

- Operating System - Windows 7,8,10
- Programming Language – Python
- Framework – Pycharm

3.2.2.1 Library Requirements

We will be using the following libraries to complete our classification problem:

OS :

- It is a module in Python provides functions for creating and removing a directory (folder), fetching its contents, changing and identifying the current directory, etc.
- Imported as : *import os*

Sys:

- It provides various functions and variables that are used to manipulate different parts of the Python runtime environment.
- Imported as : *import sys*

Flask:

- Flask is a web application framework written in Python. Flask is based on the Werkzeug WSGI toolkit and Jinja2 template engine. Both are Pocco projects.
- Imported as : *from flask import Flask*

Bootstrap:

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains HTML, CSS and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

Python

Python is a high-level, interpreted, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is dynamically-typed and garbage-collected.

HTML

The Hyper Text Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser

CSS

Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript

CHAPTER 4

SYSTEM DESIGN

4.1 System Architecture

A system architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

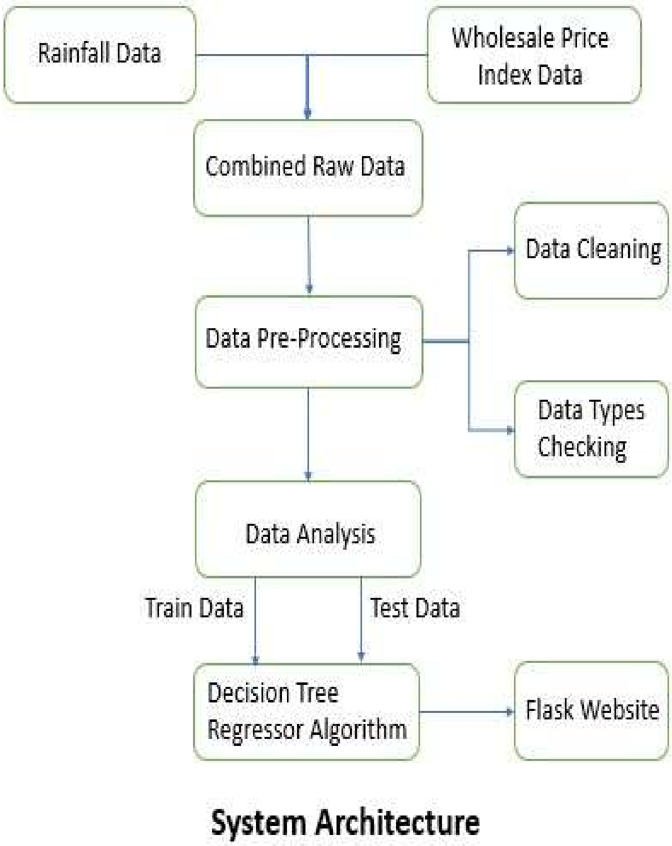


Fig 4.1: System Architecture

4.2 Training Process

Training and Testing:

- The dataset is split into two sets, training dataset and testing dataset. 80% for training the dataset & 20% for testing the dataset.
- The training dataset is used to train and fit the model while the test dataset is used to test the accuracy on an algorithm of trained model.

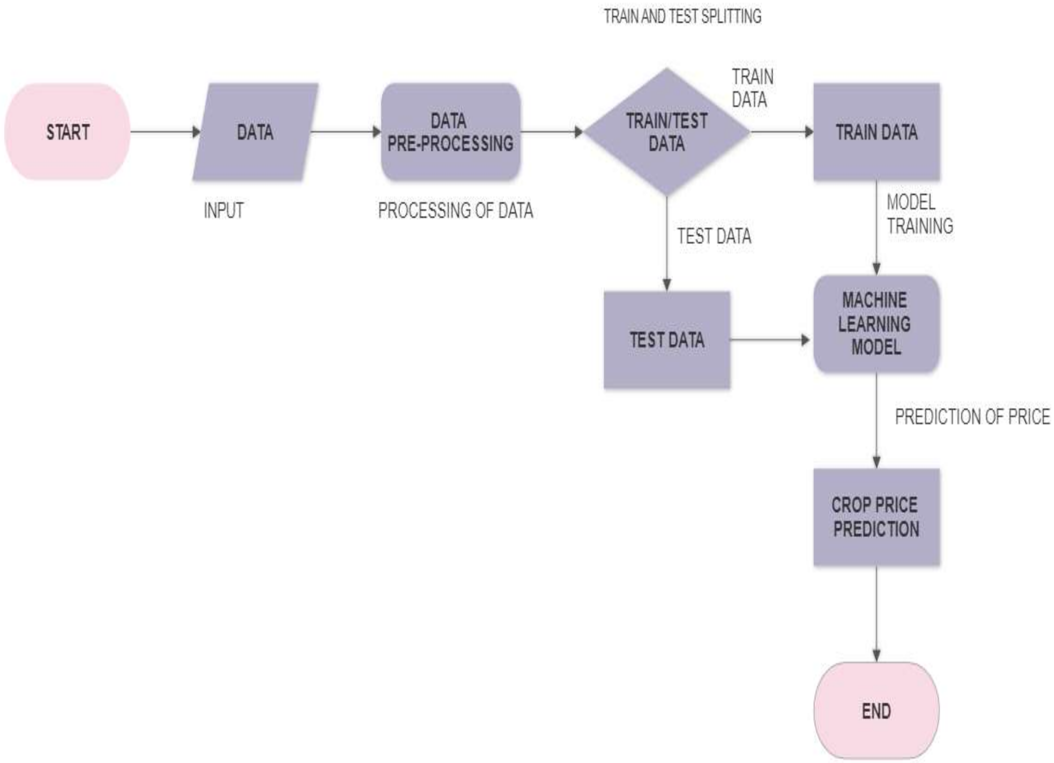


Fig 4.2: FLOW CHART

1.User Requests Prediction From Web Page

Firstly, when the user opens the website ,they can find the different crops on the list. By clicking on the respective crop image they can view the predictive price trends.

2.Retrive Current Counts From The Database

Once the user clicks on the particular crop image, the predicted values are retrived from the database and is displayed on the screen.

3.Convert The Counts From The DB into Training Model

Behind the scene the entire dataset is given as input to the training model(algorithm) and henceforth is displayed in an efficient way on the webpage.

4.Run Prediction Engine Based On Training Model

Different combination of values is being tested and trained to get accurate and precised results.

5.Retrive Predicted Counts To The Web App

After the successful testing and training of the model , the results are furnished on the webpage.

6.Display The Counts On The Web Page

Finally , an interactive website is created which can be easily accessed by any individual.

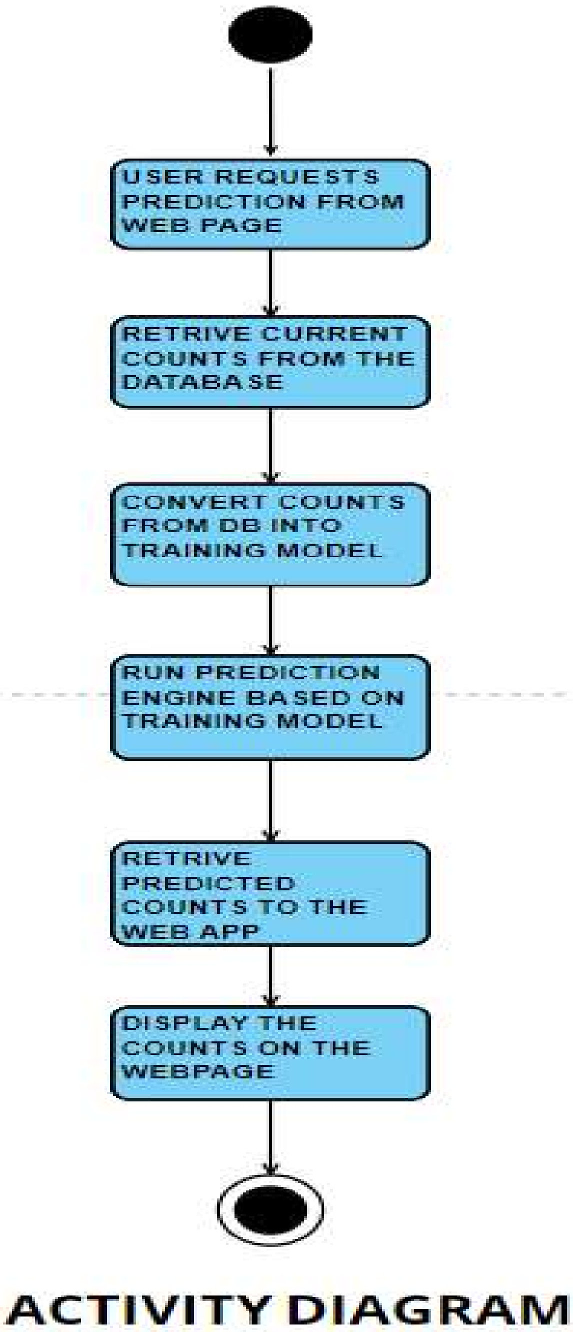


Fig 4.3: ACTIVITY DIAGRAM

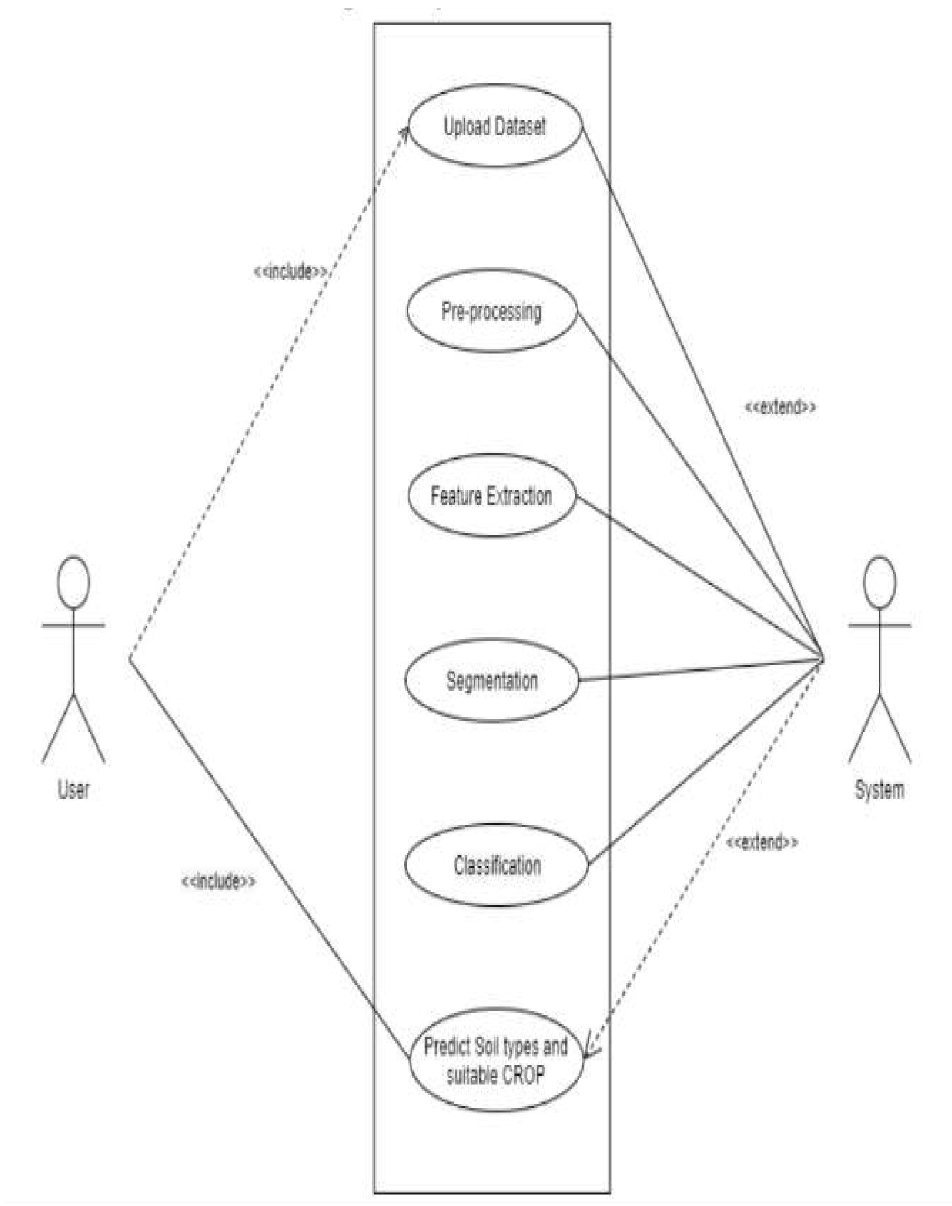


Fig 4.4: USE-CASE DIAGRAM

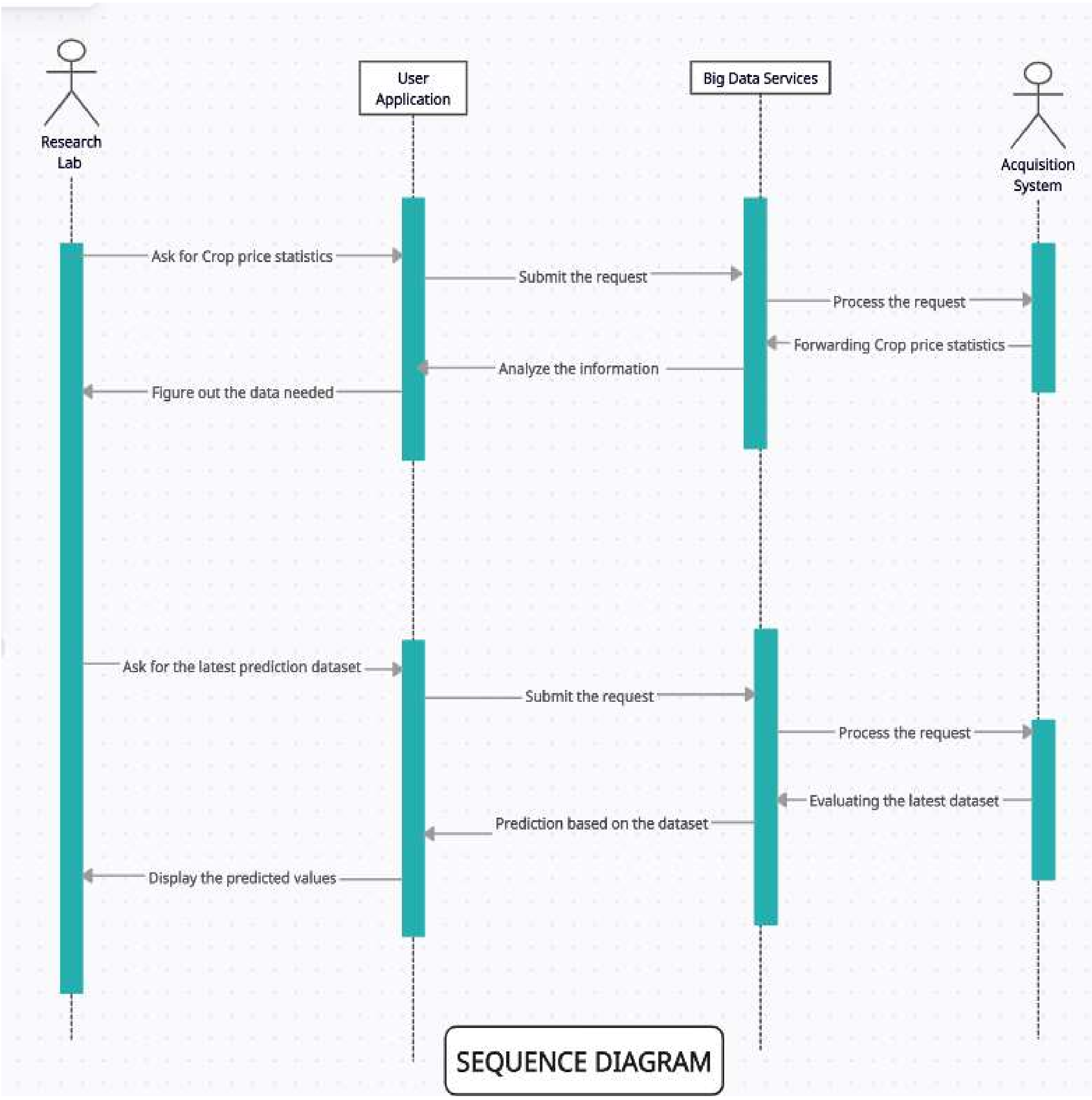


Fig 4.5: <u>SEQUENCE DIAGRAM</u>

- UML class diagram of the concept of prediction techniques and related classes.
- Classes in blue are part of procedural knowledge and classes in red are part of expert knowledge.

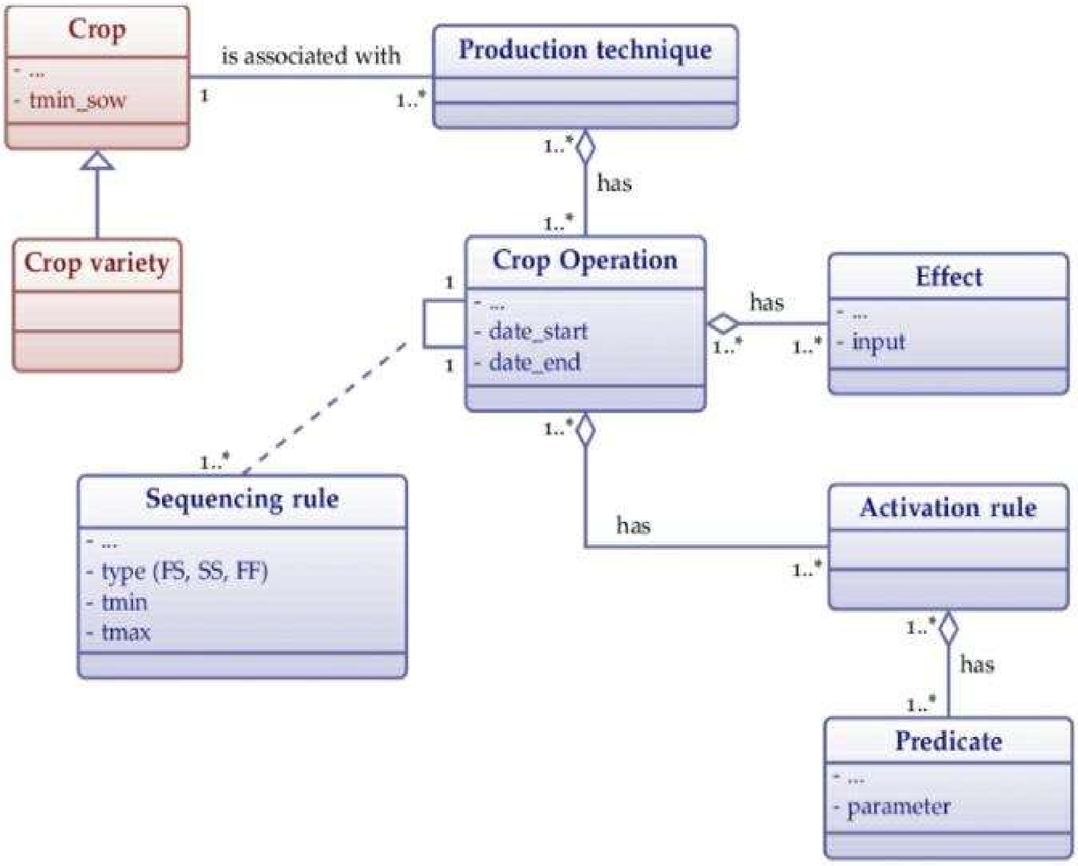


Fig 4.6: <u>UML DIAGRAM</u>

CHAPTER 5

IMPLEMENTATION

5.1 Front End And Backend Development:

5.1.1 Data.gov.in

Open Government Data Platform India or data.gov.in is a platform for supporting Open data initiative of Government of India. This portal is a single-point access to datasets, documents, services, tools and applications published by ministries, departments and organisations of the Government of India.

The site is based on Drupal Framework, and has four major modules:

- Data Management System (DMS): This facilitates publishing of datasets/applications by authorised users from Ministries/Departments/Organisations.
- Content Management System (CMS): This module is used to update or create content and functionalities for Data Portal India.
- Visitor Relationship Management (VRM): This module facilitates collation and dissemination of feedback/suggestions received on Data Portal India.
- Communities: People with specific interest can connect through online communities.

The product is developed based on the Open Government platform and its source code is available on GitHub.

5.1.2 UI/UX Interface

User experience design is the process of defining the experience a user would go through when interacting with a company, its services, and its products. Design decisions in UX design are often driven by research, data analysis, and test results rather than aesthetic preferences and opinions. Here are six programming languages UX designers will most likely encounter during their careers.

- HTML
- CSS
- Javascript
- Java
- PHP
- Python

5.1.3 Backend Development

Flask is a web application framework written in Python. It is developed by Armin Ronacher, who leads an international group of Python enthusiasts named Pocco. Flask is based on the Werkzeug WSGI toolkit and Jinja2 template engine. Both are Pocco projects. Web Server Gateway Interface (WSGI) has been adopted as a standard for Python web application development. WSGI is a specification for a universal interface between the web server and the web applications. It is a WSGI toolkit, which implements requests, response objects, and other utility functions. This enables building a web framework on top of it. The Flask framework uses Werkzeug as one of its bases. Jinja2 is a popular templating engine for Python. A web templating system combines a template with a certain data source to render dynamic web pages.Flask is often referred to as a micro framework. It aims to keep the core of an application simple yet extensible. Flask does not have built-in abstraction layer for database handling, nor does it have form a validation support. Instead, Flask supports the extensions to add such functionality to the application.

5.1.4 Machine Learning Model

A Decision Tree is a predictive approach in ML to determine what class an object belongs to. As the name suggests, a decision tree is a tree-like flow chart where the class of an object is determined step-by-step using certain known conditions. Regression in data science and machine learning is a statistical method that enables predicting outcomes based on a set of input variables. The outcome is often a variable that depends on a combination of the input variables.

5.1.5 Training Data

Training data is the data you use to train a machine learning algorithm or model to accurately predict a particular outcome, or answer, that you want your model to predict. The training data varies depending on whether we are using Supervised Learning or Unsupervised Learning Algorithms.

5.1.6 Testing Data

Once we train the model with the training dataset, it's time to test the model with the test dataset. This dataset evaluates the performance of the model and ensures that the model can generalize well with the new or unseen dataset. The test dataset is another subset of original data, which is independent of the training dataset. However, it has some similar types of features and class probability distribution and uses it as a benchmark for model evaluation once the model training is completed. Test data is a well-organized dataset that contains data for each type of scenario for a given problem that the model would be facing when used in the real world. Usually, the test dataset is approximately 20-25% of the total original data for an ML project.

At this stage, we can also check and compare the testing accuracy with the training accuracy, which means how accurate our model is with the test dataset against the training dataset. If the accuracy of the model on training data is greater than that on testing data, then the model is said to have overfitting.

The testing data should:

- Represent or part of the original dataset.
- It should be large enough to give meaningful predictions.

5.1.7 Entropy Level

Entropy is the measurement of disorder or impurities in the information processed in machine learning. It determines how a decision tree chooses to split data.Based on the entropy level(high / low) the decision /prediction is made.

5.1.8 Web Browser

Any web browser fits the best for our web application. Any version of the web browser works fine for the execution of the application.There are no much specifications needed to run this web app.

5.2 SOURCE CODE:

App.py

```
from flask import Flask, render_template
from flask_cors import CORS, cross_origin
import numpy as np
import pandas as pd
from datetime import datetime
import crops
import random

# import matplotlib.pyplot as plt

app = Flask(__name__)
app.config['CORS_HEADERS'] = 'Content-Type'

cors = CORS(app, resources={r"/ticker": {"origins": "http://localhost:port"}})

commodity_dict = {
    "arhar": "static/Arhar.csv",
    "bajra": "static/Bajra.csv",
    "barley": "static/Barley.csv",
    "copra": "static/Copra.csv",
    "cotton": "static/Cotton.csv",
    "sesamum": "static/Sesamum.csv",
    "gram": "static/Gram.csv",
    "groundnut": "static/Groundnut.csv",
    "jowar": "static/Jowar.csv",
    "maize": "static/Maize.csv",
    "masoor": "static/Masoor.csv",
    "moong": "static/Moong.csv",
    "niger": "static/Niger.csv",
    "paddy": "static/Paddy.csv",
    "ragi": "static/Ragi.csv",
    "rape": "static/Rape.csv",
    "jute": "static/Jute.csv",
  "safflower": "static/Safflower.csv",
    "soyabean": "static/Soyabean.csv",
    "sugarcane": "static/Sugarcane.csv",
    "sunflower": "static/Sunflower.csv",
    "urad": "static/Urad.csv",
    "wheat": "static/Wheat.csv"
}

annual_rainfall = [14.8, 7.2, 37.6, 41.4, 67.5, 148.6, 315.9, 162.7, 190.0, 50.8, 34.6, 25.5]
base = {
    "Paddy": 2183,
```

```
    "Arhar": 7000,
    "Bajra": 2500,
    "Barley": 1850,
    "Copra": 10860,
    "Cotton": 6620,
    "Sesamum": 8635,
    "Gram": 5440,
    "Groundnut": 6377,
    "Jowar": 3225,
    "Maize": 2090,
    "Masoor": 6425,
    "Moong": 8558,
    "Niger": 7734,
    "Ragi": 3846,
    "Rape": 5650,
    "Jute": 5050,
    "Safflower": 5800,
    "Soyabean": 4600,
    "Sugarcane": 3050,
    "Sunflower": 6760,
    "Urad": 6950,
    "Wheat": 2275

}
commodity_list = []
class Commodity:

    def __init__(self, csv_name):
        self.name = csv_name
        dataset = pd.read_csv(csv_name)
        self.X = dataset.iloc[:, :-1].values
        self.Y = dataset.iloc[:, 3].values

        #from sklearn.model_selection import train_test_split
        #X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.1,
random_state=0)

        # Fitting decision tree regression to dataset
        from sklearn.tree import DecisionTreeRegressor
        depth = random.randrange(7,18)
        self.regressor = DecisionTreeRegressor(max_depth=depth)
        self.regressor.fit(self.X, self.Y)
        #y_pred_tree = self.regressor.predict(X_test)
        # fsa=np.array([float(1),2019,45]).reshape(1,3)
        # fask=regressor_tree.predict(fsa)

    def getPredictedValue(self, value):
        if value[1]>=2019:
            fsa = np.array(value).reshape(1, 3)
            #print(" ",self.regressor.predict(fsa)[0])
```

```
            return self.regressor.predict(fsa)[0]
        else:
            c=self.X[:,0:2]
            x=[]
            for i in c:
                x.append(i.tolist())
            fsa = [value[0], value[1]]
            ind = 0
            for i in range(0,len(x)):
                if x[i]==fsa:
                    ind=i
                    break
            #print(index, " ",ind)
            #print(x[ind])
            #print(self.Y[i])
            return self.Y[i]

    def getCropName(self):
        a = self.name.split('.')
        return a[0]

@app.route('/')
def index():
    context = {
        "top5": TopFiveWinners(),
        "bottom5": TopFiveLosers(),
        "sixmonths": SixMonthsForecast()
    }
    return render_template('index.html', context=context)


@app.route('/commodity/<name>')
def crop_profile(name):
    max_crop, min_crop, forecast_crop_values = TwelveMonthsForecast(name)
    prev_crop_values = TwelveMonthPrevious(name)
    forecast_x = [i[0] for i in forecast_crop_values]
    forecast_y = [i[1] for i in forecast_crop_values]
    previous_x = [i[0] for i in prev_crop_values]
    previous_y = [i[1] for i in prev_crop_values]
    current_price = CurrentMonth(name)
    #print(max_crop)
    #print(min_crop)
    #print(forecast_crop_values)
    #print(prev_crop_values)
    #print(str(forecast_x))
    crop_data = crops.crop(name)
    context = {
        "name":name,
        "max_crop": max_crop,
        "min_crop": min_crop,
```

```
        "forecast_values": forecast_crop_values,
        "forecast_x": str(forecast_x),
        "forecast_y":forecast_y,
        "previous_values": prev_crop_values,
        "previous_x":previous_x,
        "previous_y":previous_y,
        "current_price": current_price,
        "image_url":crop_data[0],
        "prime_loc":crop_data[1],
        "type_c":crop_data[2],
        "export":crop_data[3]
    }
    return render_template('commodity.html', context=context)

@app.route('/ticker/<item>/<number>')
@cross_origin(origin='localhost',headers=['Content- Type','Authorization'])
def ticker(item, number):
    n = int(number)
    i = int(item)
    data = SixMonthsForecast()
    context = str(data[n][i])

    if i == 2 or i == 5:
        context = '₹' + context
    elif i == 3 or i == 6:

        context = context + '%'

    #print('context: ', context)
    return context


def TopFiveWinners():
    current_month = datetime.now().month
    current_year = datetime.now().year
    current_rainfall = annual_rainfall[current_month - 1]
    prev_month = current_month - 1
    prev_rainfall = annual_rainfall[prev_month - 1]
    current_month_prediction = []
    prev_month_prediction = []
change = []

        for i in commodity_list:
        current_predict = i.getPredictedValue([float(current_month), current_year,
current_rainfall])
        current_month_prediction.append(current_predict)
        prev_predict = i.getPredictedValue([float(prev_month), current_year, prev_rainfall])
        prev_month_prediction.append(prev_predict)
change.append((((current_predict - prev_predict) * 100 / prev_predict),
commodity_list.index(i)))
```

```
sorted_change = change
sorted_change.sort(reverse=True)
# print(sorted_change)
to_send = []
for j in range(0, 5):
perc, i = sorted_change[j]
name = commodity_list[i].getCropName().split('/')[1]
to_send.append([name, round((current_month_prediction[i] * base[name]) / 100, 2),
round(perc, 2)])
#print(to_send)
return to_send


def TopFiveLosers():
current_month = datetime.now().month
current_year = datetime.now().year
current_rainfall = annual_rainfall[current_month - 1]
prev_month = current_month - 1
prev_rainfall = annual_rainfall[prev_month - 1]
current_month_prediction = []
prev_month_prediction = []
change = []

for i in commodity_list:
current_predict = i.getPredictedValue([float(current_month), current_year,
current_rainfall])
current_month_prediction.append(current_predict)
prev_predict = i.getPredictedValue([float(prev_month), current_year, prev_rainfall])
prev_month_prediction.append(prev_predict)
change.append((((current_predict - prev_predict) * 100 / prev_predict),
commodity_list.index(i)))
sorted_change = change
sorted_change.sort()
to_send = []
for j in range(0, 5):
perc, i = sorted_change[j]
name = commodity_list[i].getCropName().split('/')[1]
to_send.append([name, round((current_month_prediction[i] * base[name]) / 100, 2),
round(perc, 2)])
# print(to_send)
return to_send



def SixMonthsForecast():
month1=[]
month2=[]
month3=[]
month4=[]
   month5=[]
```

```
    month6=[]
    for i in commodity_list:
        crop=SixMonthsForecastHelper(i.getCropName())
        k=0
        for j in crop:
            time = j[0]
            price = j[1]
            change = j[2]
            if k==0:
                month1.append((price,change,i.getCropName().split("/")[1],time))
            elif k==1:
                month2.append((price,change,i.getCropName().split("/")[1],time))
            elif k==2:
                month3.append((price,change,i.getCropName().split("/")[1],time))
            elif k==3:
                month4.append((price,change,i.getCropName().split("/")[1],time))
            elif k==4:
                month5.append((price,change,i.getCropName().split("/")[1],time))
            elif k==5:
                month6.append((price,change,i.getCropName().split("/")[1],time))
            k+=1
    month1.sort()
    month2.sort()
    month3.sort()
    month4.sort()
    month5.sort()
    month6.sort()
    crop_month_wise=[]
    crop_month_wise.append([month1[0][3],month1[len(month1)-1]
[2],month1[len(month1)-1][0],month1[len(month1)-1][1],month1[0][2],month1[0]
[0],month1[0][1]])
    crop_month_wise.append([month2[0][3],month2[len(month2)-1]
[2],month2[len(month2)-1][0],month2[len(month2)-1][1],month2[0][2],month2[0]
[0],month2[0][1]])
    crop_month_wise.append([month3[0][3],month3[len(month3)-1]
[2],month3[len(month3)-1][0],month3[len(month3)-1][1],month3[0][2],month3[0]
[0],month3[0][1]])
    crop_month_wise.append([month4[0][3],month4[len(month4)-1]
[2],month4[len(month4)-1][0],month4[len(month4)-1][1],month4[0][2],month4[0]
[0],month4[0][1]])
    crop_month_wise.append([month5[0][3],month5[len(month5)-1]
[2],month5[len(month5)-1][0],month5[len(month5)-1][1],month5[0][2],month5[0]
[0],month5[0][1]])
    crop_month_wise.append([month6[0][3],month6[len(month6)-1]
[2],month6[len(month6)-1][0],month6[len(month6)-1][1],month6[0][2],month6[0]
[0],month6[0][1]])

   # print(crop_month_wise)
    return crop_month_wise
```

```
def SixMonthsForecastHelper(name):
    current_month = datetime.now().month
    current_year = datetime.now().year
    current_rainfall = annual_rainfall[current_month - 1]
    name = name.split("/")[1]
    name = name.lower()
    commodity = commodity_list[0]
    for i in commodity_list:
        if name == str(i):
            commodity = i
            break
    month_with_year = []
    for i in range(1, 7):
        if current_month + i <= 12:
            month_with_year.append((current_month + i, current_year,
annual_rainfall[current_month + i - 1]))
        else:
            month_with_year.append((current_month + i - 12, current_year + 1,
annual_rainfall[current_month + i - 13]))
  wpis = []
    current_wpi = commodity.getPredictedValue([float(current_month), current_year,
current_rainfall])
    change = []

    for m, y, r in month_with_year:
        current_predict = commodity.getPredictedValue([float(m), y, r])
        wpis.append(current_predict)
        change.append(((current_predict - current_wpi) * 100) / current_wpi)

    crop_price = []
    for i in range(0, len(wpis)):
        m, y, r = month_with_year[i]
        x = datetime(y, m, 1)
        x = x.strftime("%b %y")
        crop_price.append([x, round((wpis[i]* base[name.capitalize()]) / 100, 2) ,
round(change[i], 2)])

   # print("Crop_Price: ", crop_price)
    return crop_price

def CurrentMonth(name):
    current_month = datetime.now().month
    current_year = datetime.now().year
    current_rainfall = annual_rainfall[current_month - 1]
    name = name.lower()
    commodity = commodity_list[0]
    for i in commodity_list:
        if name == str(i):
            commodity = i
            break
```

```
    current_wpi = commodity.getPredictedValue([float(current_month), current_year,
current_rainfall])
    current_price = (base[name.capitalize()]*current_wpi)/100
    return current_price

def TwelveMonthsForecast(name):
    current_month = datetime.now().month
    current_year = datetime.now().year
    current_rainfall = annual_rainfall[current_month - 1]
    name = name.lower()
    commodity = commodity_list[0]
    for i in commodity_list:
        if name == str(i):
            commodity = i
            break
    month_with_year = []
    for i in range(1, 13):
        if current_month + i <= 12:
            month_with_year.append((current_month + i, current_year,
annual_rainfall[current_month + i - 1]))
        else:
            month_with_year.append((current_month + i - 12, current_year + 1,
annual_rainfall[current_month + i - 13]))
    max_index = 0
    min_index = 0
    max_value = 0
    min_value = 9999
    wpis = []
    current_wpi = commodity.getPredictedValue([float(current_month), current_year,
current_rainfall])
    change = []

    for m, y, r in month_with_year:
        current_predict = commodity.getPredictedValue([float(m), y, r])
        if current_predict > max_value:
            max_value = current_predict
            max_index = month_with_year.index((m, y, r))
        if current_predict < min_value:
            min_value = current_predict
            min_index = month_with_year.index((m, y, r))
        wpis.append(current_predict)
        change.append(((current_predict - current_wpi) * 100) / current_wpi)

    max_month, max_year, r1 = month_with_year[max_index]
    min_month, min_year, r2 = month_with_year[min_index]
    min_value = min_value * base[name.capitalize()] / 100
    max_value = max_value * base[name.capitalize()] / 100
    crop_price = []
    for i in range(0, len(wpis)):
        m, y, r = month_with_year[i]
```

```
        x = datetime(y, m, 1)
        x = x.strftime("%b %y")
        crop_price.append([x, round((wpis[i]* base[name.capitalize()]) / 100, 2) ,
round(change[i], 2)])
   # print("forecasr", wpis)
    x = datetime(max_year,max_month,1)
    x = x.strftime("%b %y")
    max_crop = [x, round(max_value,2)]
    x = datetime(min_year, min_month, 1)
    x = x.strftime("%b %y")
    min_crop = [x, round(min_value,2)]

    return max_crop, min_crop, crop_price


def TwelveMonthPrevious(name):
    name = name.lower()
    current_month = datetime.now().month
    current_year = datetime.now().year
    current_rainfall = annual_rainfall[current_month - 1]
    commodity = commodity_list[0]
    wpis = []
    crop_price = []
    for i in commodity_list:
        if name == str(i):
            commodity = i
            break
    month_with_year = []
    for i in range(1, 13):
        if current_month - i >= 1:
            month_with_year.append((current_month - i, current_year,
annual_rainfall[current_month - i - 1]))
        else:
            month_with_year.append((current_month - i + 12, current_year - 1,
annual_rainfall[current_month - i + 11]))

    for m, y, r in month_with_year:
        current_predict = commodity.getPredictedValue([float(m), 2013, r])
        wpis.append(current_predict)

    for i in range(0, len(wpis)):
        m, y, r = month_with_year[i]
        x = datetime(y,m,1)
        x = x.strftime("%b %y")
        crop_price.append([x, round((wpis[i]* base[name.capitalize()]) / 100, 2)])
   # print("previous ", wpis)
    new_crop_price =[]
    for i in range(len(crop_price)-1,-1,-1):
        new_crop_price.append(crop_price[i])
    return new_crop_price
```

```
if __name__ == "__main__":
    arhar = Commodity(commodity_dict["arhar"])
    commodity_list.append(arhar)
    bajra = Commodity(commodity_dict["bajra"])
    commodity_list.append(bajra)
    barley = Commodity(commodity_dict["barley"])
    commodity_list.append(barley)
    copra = Commodity(commodity_dict["copra"])
    commodity_list.append(copra)
    cotton = Commodity(commodity_dict["cotton"])
    commodity_list.append(cotton)
    sesamum = Commodity(commodity_dict["sesamum"])
    commodity_list.append(sesamum)
    gram = Commodity(commodity_dict["gram"])
    commodity_list.append(gram)
    groundnut = Commodity(commodity_dict["groundnut"])
    commodity_list.append(groundnut)
    jowar = Commodity(commodity_dict["jowar"])
    commodity_list.append(jowar)
    maize = Commodity(commodity_dict["maize"])
    commodity_list.append(maize)
    masoor = Commodity(commodity_dict["masoor"])
    commodity_list.append(masoor)
    moong = Commodity(commodity_dict["moong"])
    commodity_list.append(moong)
    niger = Commodity(commodity_dict["niger"])
    commodity_list.append(niger)
    paddy = Commodity(commodity_dict["paddy"])
    commodity_list.append(paddy)
    ragi = Commodity(commodity_dict["ragi"])
    commodity_list.append(ragi)
    rape = Commodity(commodity_dict["rape"])
    commodity_list.append(rape)
    jute = Commodity(commodity_dict["jute"])
    commodity_list.append(jute)
    safflower = Commodity(commodity_dict["safflower"])
    commodity_list.append(safflower)
    soyabean = Commodity(commodity_dict["soyabean"])
    commodity_list.append(soyabean)
    sugarcane = Commodity(commodity_dict["sugarcane"])
    commodity_list.append(sugarcane)
    sunflower = Commodity(commodity_dict["sunflower"])
    commodity_list.append(sunflower)
    urad = Commodity(commodity_dict["urad"])
    commodity_list.append(urad)
    wheat = Commodity(commodity_dict["wheat"])
    commodity_list.append(wheat)
```

```
app.run()
```

workspace.xml


```
<?xml version="1.0" encoding="UTF-8"?>
<project version="4">
  <component name="ChangeListManager">
    <list default="true" id="605f3263-2147-4e0b-bd71-956920c2fdd3" name="Default"
comment="">
      <change beforePath="$PROJECT_DIR$/.idea/workspace.xml" beforeDir="false"
afterPath="$PROJECT_DIR$/.idea/workspace.xml" afterDir="false" />
      <change beforePath="$PROJECT_DIR$/app.py" beforeDir="false"
afterPath="$PROJECT_DIR$/app.py" afterDir="false" />
      <change beforePath="$PROJECT_DIR$/templates/index.html" beforeDir="false"
afterPath="$PROJECT_DIR$/templates/index.html" afterDir="false" />
    </list>
    <option name="EXCLUDED_CONVERTED_TO_IGNORED" value="true" />
    <option name="TRACKING_ENABLED" value="true" />
    <option name="SHOW_DIALOG" value="false" />
    <option name="HIGHLIGHT_CONFLICTS" value="true" />
    <option name="HIGHLIGHT_NON_ACTIVE_CHANGELIST" value="false" />
    <option name="LAST_RESOLUTION" value="IGNORE" />
  </component>
  <component name="FileEditorManager">
    <leaf SIDE_TABS_SIZE_LIMIT_KEY="300">
      <file leaf-file-name="crops.py" pinned="false" current-in-tab="false">
        <entry file="file://$PROJECT_DIR$/crops.py">
          <provider selected="true" editor-type-id="text-editor">
            <state relative-caret-position="25">
              <caret line="8" column="87" selection-start-line="8" selection-start-
column="87" selection-end-line="8" selection-end-column="87" />
            </state>
          </provider>
        </entry>
      </file>
      <file leaf-file-name="app.py" pinned="false" current-in-tab="false">
        <entry file="file://$PROJECT_DIR$/app.py">
          <provider selected="true" editor-type-id="text-editor">
            <state relative-caret-position="406">
              <caret line="318" column="4" selection-start-line="318" selection-start-
column="4" selection-end-line="318" selection-end-column="4" />
              <folding>
                <element signature="e#109#150#0" expanded="true" />
              </folding>
            </state>
          </provider>
        </entry>
      </file>
     <file leaf-file-name="commodity.html" pinned="false" current-in-tab="false">
```

```
      <entry file="file://$PROJECT_DIR$/templates/commodity.html">
        <provider selected="true" editor-type-id="text-editor">
          <state relative-caret-position="225">
            <caret line="37" column="25" selection-start-line="37" selection-start-
column="25" selection-end-line="37" selection-end-column="25" />
          </state>
        </provider>
      </entry>
    </file>
    <file leaf-file-name="index.html" pinned="false" current-in-tab="true">
      <entry file="file://$PROJECT_DIR$/templates/index.html">
        <provider selected="true" editor-type-id="text-editor">
          <state relative-caret-position="110">
            <caret line="109" column="94" selection-start-line="109" selection-start-
column="94" selection-end-line="109" selection-end-column="94" />
            <folding>
              <element
signature="n#style#0;n#img#0;n#p#0;n#td#2;n#tr#0;n#table#0;n#div#0;n#div#0;n#div#1
;n#div#1;n#div#0;n#body#0;n#html#0;n#!!top" expanded="true" />
              <element
signature="n#style#0;n#img#0;n#p#0;n#td#2;n#tr#1;n#table#0;n#div#0;n#div#0;n#div#1
;n#div#1;n#div#0;n#body#0;n#html#0;n#!!top" expanded="true" />
              <element
signature="n#style#0;n#a#0;n#div#0;n#div#2;n#div#0;n#body#0;n#html#0;n#!!top"
expanded="true" />
              <element
signature="n#style#0;n#a#0;n#div#1;n#div#2;n#div#0;n#body#0;n#html#0;n#!!top"
expanded="true" />
              <element
signature="n#style#0;n#a#0;n#div#2;n#div#2;n#div#0;n#body#0;n#html#0;n#!!top"
expanded="true" />
              <element
signature="n#style#0;n#a#0;n#div#3;n#div#2;n#div#0;n#body#0;n#html#0;n#!!top"
expanded="true" />
              <element
signature="n#style#0;n#a#0;n#div#0;n#div#3;n#div#0;n#body#0;n#html#0;n#!!top"
expanded="true" />
              <element
signature="n#style#0;n#a#0;n#div#1;n#div#3;n#div#0;n#body#0;n#html#0;n#!!top"
expanded="true" />
              <element
signature="n#style#0;n#a#0;n#div#2;n#div#3;n#div#0;n#body#0;n#html#0;n#!!top"
expanded="true" />
              <element
signature="n#style#0;n#a#0;n#div#3;n#div#3;n#div#0;n#body#0;n#html#0;n#!!top"
expanded="true" />
              <element
signature="n#style#0;n#a#0;n#div#0;n#div#4;n#div#0;n#body#0;n#html#0;n#!!top"
expanded="true" />
```

```
        <element
signature="n#style#0;n#a#0;n#div#1;n#div#4;n#div#0;n#body#0;n#html#0;n#!!top"
expanded="true" />
        <element
signature="n#style#0;n#a#0;n#div#2;n#div#4;n#div#0;n#body#0;n#html#0;n#!!top"
expanded="true" />
        <element
signature="n#style#0;n#a#0;n#div#3;n#div#4;n#div#0;n#body#0;n#html#0;n#!!top"
expanded="true" />
        <element
signature="n#style#0;n#a#0;n#div#0;n#div#5;n#div#0;n#body#0;n#html#0;n#!!top"
expanded="true" />
        <element
signature="n#style#0;n#a#0;n#div#1;n#div#5;n#div#0;n#body#0;n#html#0;n#!!top"
expanded="true" />
        <element
signature="n#style#0;n#a#0;n#div#2;n#div#5;n#div#0;n#body#0;n#html#0;n#!!top"
expanded="true" />
        <element
signature="n#style#0;n#a#0;n#div#3;n#div#5;n#div#0;n#body#0;n#html#0;n#!!top"
expanded="true" />
        <element
signature="n#style#0;n#a#0;n#div#0;n#div#6;n#div#0;n#body#0;n#html#0;n#!!top"
expanded="true" />
        <element
signature="n#style#0;n#a#0;n#div#1;n#div#6;n#div#0;n#body#0;n#html#0;n#!!top"
expanded="true" />
        <element
signature="n#style#0;n#a#0;n#div#2;n#div#6;n#div#0;n#body#0;n#html#0;n#!!top"
expanded="true" />
        <element
signature="n#style#0;n#a#0;n#div#3;n#div#6;n#div#0;n#body#0;n#html#0;n#!!top"
expanded="true" />
        <element
signature="n#style#0;n#a#0;n#div#0;n#div#7;n#div#0;n#body#0;n#html#0;n#!!top"
expanded="true" />
        <element
signature="n#style#0;n#a#0;n#div#1;n#div#7;n#div#0;n#body#0;n#html#0;n#!!top"
expanded="true" />
        <element
signature="n#style#0;n#a#0;n#div#2;n#div#7;n#div#0;n#body#0;n#html#0;n#!!top"
expanded="true" />
       </folding>
      </state>
     </provider>
    </entry>
   </file>
  </leaf>
 </component>
 <component name="FileTemplateManagerImpl">
```

```
    <option name="RECENT_TEMPLATES">
      <list>
        <option value="HTML File" />
      </list>
    </option>
  </component>
  <component name="FindInProjectRecents">
    <findStrings>
      <find>__ini</find>
      <find>print</find>
    </findStrings>
  </component>
 <component name="Git.Settings">
    <option name="RECENT_GIT_ROOT_PATH" value="$PROJECT_DIR$" />
  </component>
  <component name="IdeDocumentHistory">
    <option name="CHANGED_PATHS">
      <list>
        <option value="$PROJECT_DIR$/templates/app.py" />
        <option value="$PROJECT_DIR$/templates/export" />
        <option value="$PROJECT_DIR$/crops.py" />
        <option value="$PROJECT_DIR$/templates/ticker.html" />
        <option value="$PROJECT_DIR$/templates/commodity.html" />
        <option value="$PROJECT_DIR$/app.py" />
        <option value="$PROJECT_DIR$/templates/index.html" />
      </list>
    </option>
  </component>
  <component name="ProjectFrameBounds" extendedState="6">
    <option name="x" value="-9" />
    <option name="y" value="-9" />
    <option name="width" value="977" />
    <option name="height" value="1038" />
  </component>
  <component name="ProjectView">
    <navigator proportions="" version="1">
      <foldersAlwaysOnTop value="true" />
    </navigator>
    <panes>
      <pane id="ProjectPane">
        <subPane>
          <expand>
            <path>
              <item name="Crop_Prediction" type="b2602c69:ProjectViewProjectNode" />
              <item name="Crop_Prediction" type="462c0819:PsiDirectoryNode" />
            </path>
            <path>
              <item name="Crop_Prediction" type="b2602c69:ProjectViewProjectNode" />
              <item name="Crop_Prediction" type="462c0819:PsiDirectoryNode" />
              <item name="templates" type="462c0819:PsiDirectoryNode" />
```

```
          </path>
        </expand>
        <select />


      </subPane>
      </pane>
      <pane id="Scope" />
    </panes>
  </component>
  <component name="PropertiesComponent">
    <property name="DefaultHtmlFileTemplate" value="HTML File" />
    <property name="last_opened_file_path" value="$PROJECT_DIR$" />
    <property name="settings.editor.selected.configurable"
value="com.jetbrains.python.configuration.PyActiveSdkModuleConfigurable" />
  </component>
  <component name="RunDashboard">
    <option name="ruleStates">
      <list>
        <RuleState>
          <option name="name" value="ConfigurationTypeDashboardGroupingRule" />
        </RuleState>
        <RuleState>
          <option name="name" value="StatusDashboardGroupingRule" />
        </RuleState>
      </list>
    </option>
  </component>
  <component name="RunManager">
<configuration name="app" type="PythonConfigurationType" factoryName="Python"
temporary="true">
      <module name="Crop_Prediction" />
      <option name="INTERPRETER_OPTIONS" value="" />
      <option name="PARENT_ENVS" value="true" />
      <envs>
        <env name="PYTHONUNBUFFERED" value="1" />
      </envs>
      <option name="SDK_HOME" value="" />
      <option name="WORKING_DIRECTORY" value="$PROJECT_DIR$" />
      <option name="IS_MODULE_SDK" value="true" />
      <option name="ADD_CONTENT_ROOTS" value="true" />
      <option name="ADD_SOURCE_ROOTS" value="true" />
      <option name="SCRIPT_NAME" value="$PROJECT_DIR$/app.py" />
      <option name="PARAMETERS" value="" />
      <option name="SHOW_COMMAND_LINE" value="false" />
      <option name="EMULATE_TERMINAL" value="false" />
      <option name="MODULE_MODE" value="false" />
```

CHAPTER 6

RESULTS

The proposed system ensures the following:

- The Website loads within 1 minute.
- As soon as the website loads you get directed to the homepage of the website where you can view 23 types of crops.
- Each crop is displayed using a animated icon.
- On clicking the icon you will get directed to the brief information about the crop.
- In the brief information of the any respective crop you can view the 12 months prediction along with the graphical represenation of the data.
- The graph displays the previous year and upcoming year fluctuations.
- The UI is designed in a way where the user can easily get to know about the fluctuations through graphical representations and also we can click on the respective crops through the aniamted icons.
- The crops with higher priority are displayed as Top Gainers, and the crops with lower priority are displayed as Top loosers.
- Using the entropy level the crops are classified as top fgainers and top loosers.
- In machine learning, entropy is a measure of the impurity or disorder in a set **of** data.

$$E = -\sum_{i=1}^{N} P_i \log_2 P_i$$

- Where;
  P_i = Probability of randomly selecting an example in class I;
- Entropy always lies between 0 and 1, however depending on the number of classes in the dataset, it can be greater than 1.

CHAPTER 7

SNAPSHOT

Barley			
Month	Year	Rainfall	WPI
4	2012	47.5	117.1
5	2012	31.7	118
6	2012	117.8	111.2
7	2012	250.2	113.2
8	2012	262.4	113.8
9	2012	193.5	111.8
10	2012	58.7	110.5
11	2012	30.7	113.9
12	2012	11.7	119
1	2013	11.3	119.4
2	2013	40.1	120.2
3	2013	15.7	119.5
4	2013	30.4	116.5
5	2013	57.8	114.7
6	2013	219.8	115.8
7	2013	310	115.8
8	2013	254.7	116.3
9	2013	152.7	118
10	2013	129.4	118.6
11	2013	14	120.1
12	2013	6.7	123.3

Fig 7.1: THE SAMPLE DATASET OF BARLEY

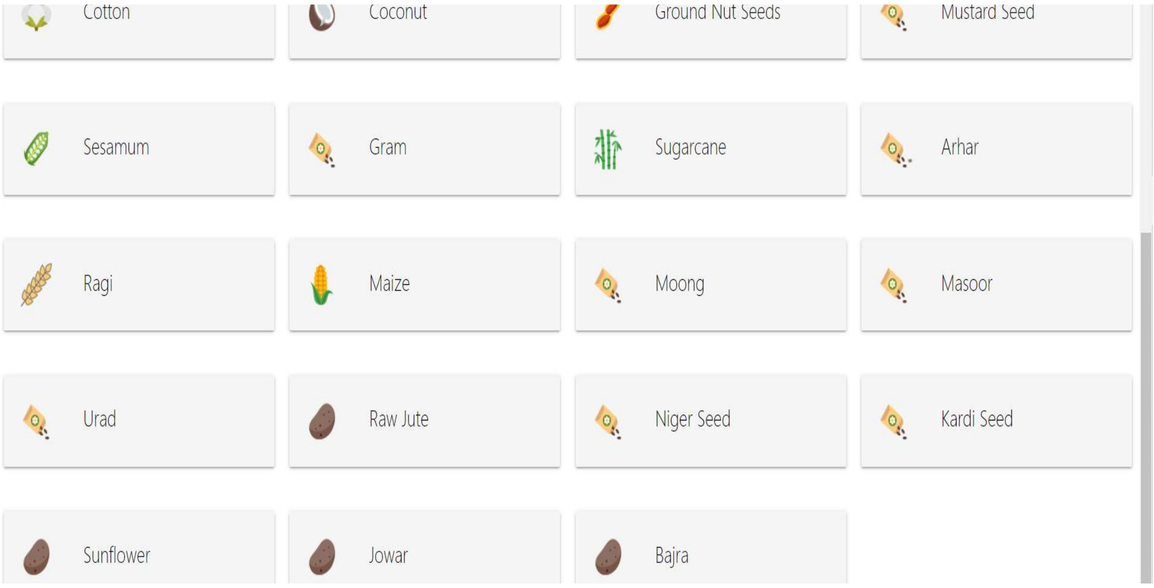


Fig 7.2: HOME PAGE OF THE WEB APPLICATION

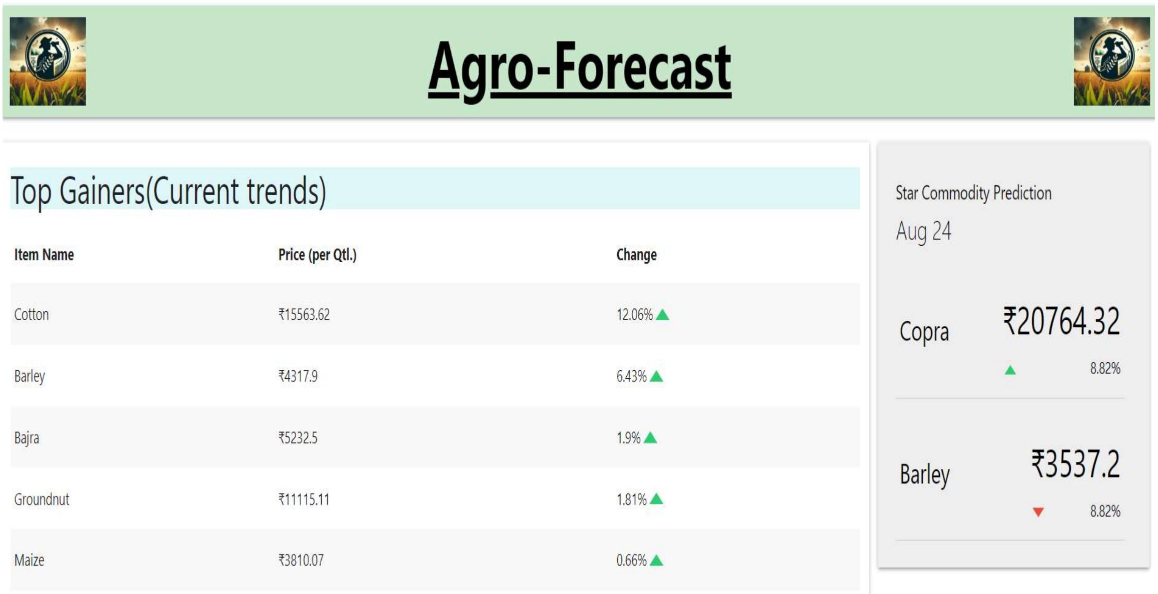


Fig 7.3: THE LIST OF TOP GAINERS

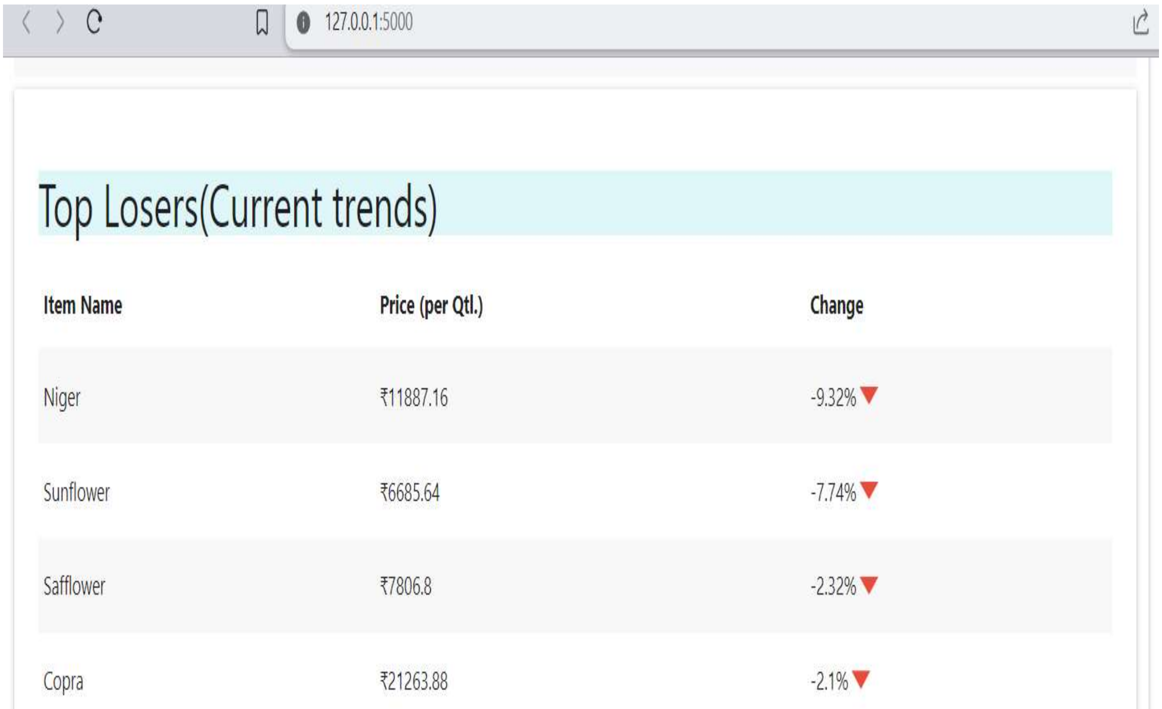


Fig 7.4: <u>THE LIST OF TOP LOOSERS</u>

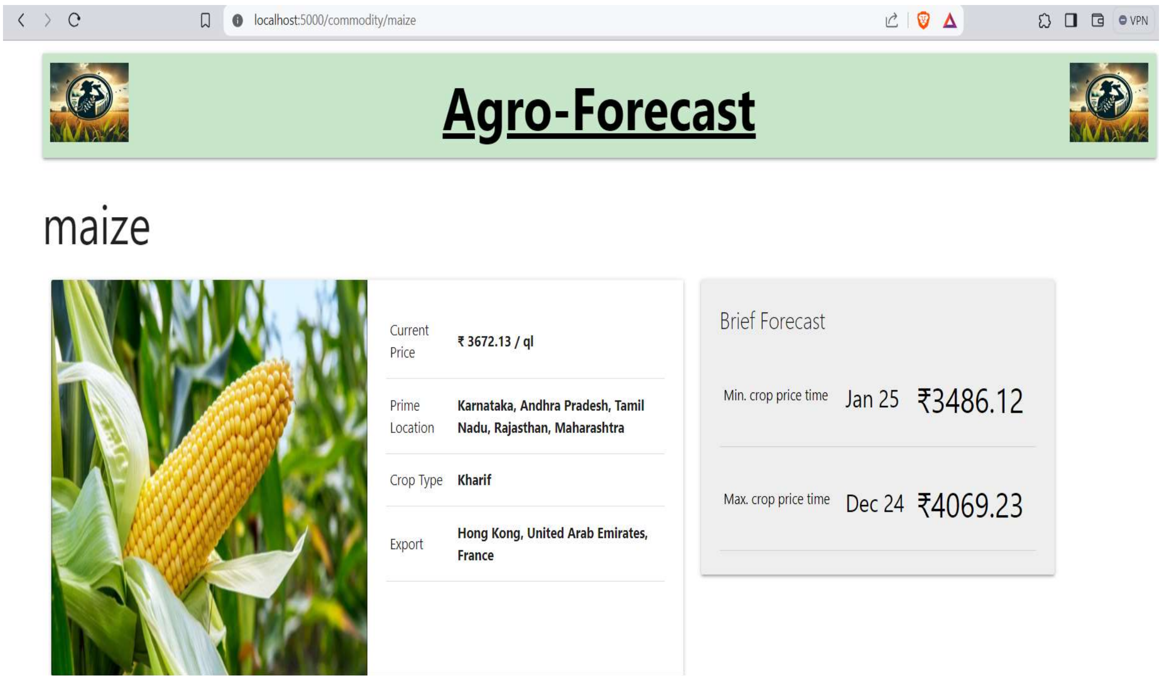
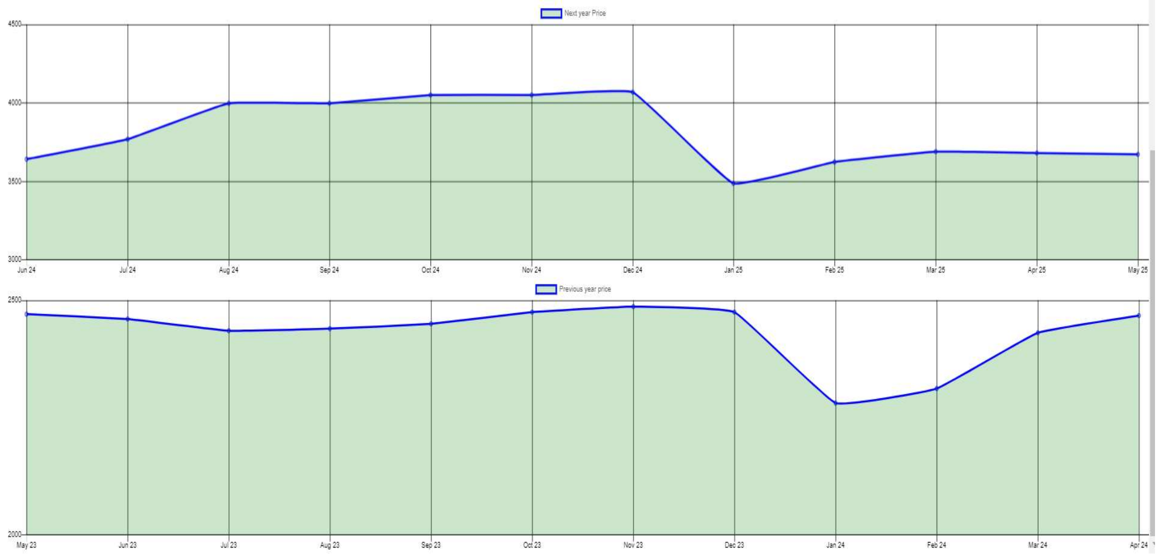


Fig 7.5: <u>BRIEF INFORMATION ON MAIZE CROP</u>



Forecast Trends

Month	Price (per Qtl.)	Change
Jun 24	₹3640.78	-0.85% ▼
Jul 24	₹3768.27	2.62% ▲
Aug 24	₹3996.08	8.82% ▲
Sep 24	₹3996.08	8.82% ▲
Oct 24	₹4048.33	10.24% ▲
Nov 24	₹4048.33	10.24% ▲
Dec 24	₹4069.23	10.81% ▲
Jan 25	₹3486.12	-5.07% ▼
Feb 25	₹3619.88	-1.42% ▼
Mar 25	₹3686.76	0.4% ▲
Apr 25	₹3680.49	0.23% ▲
May 25	₹3672.13	0.0% ▲

Fig 7.6: MAIZE CROP'S 12 MONTHS PREDICTION (WITH GRAPH)

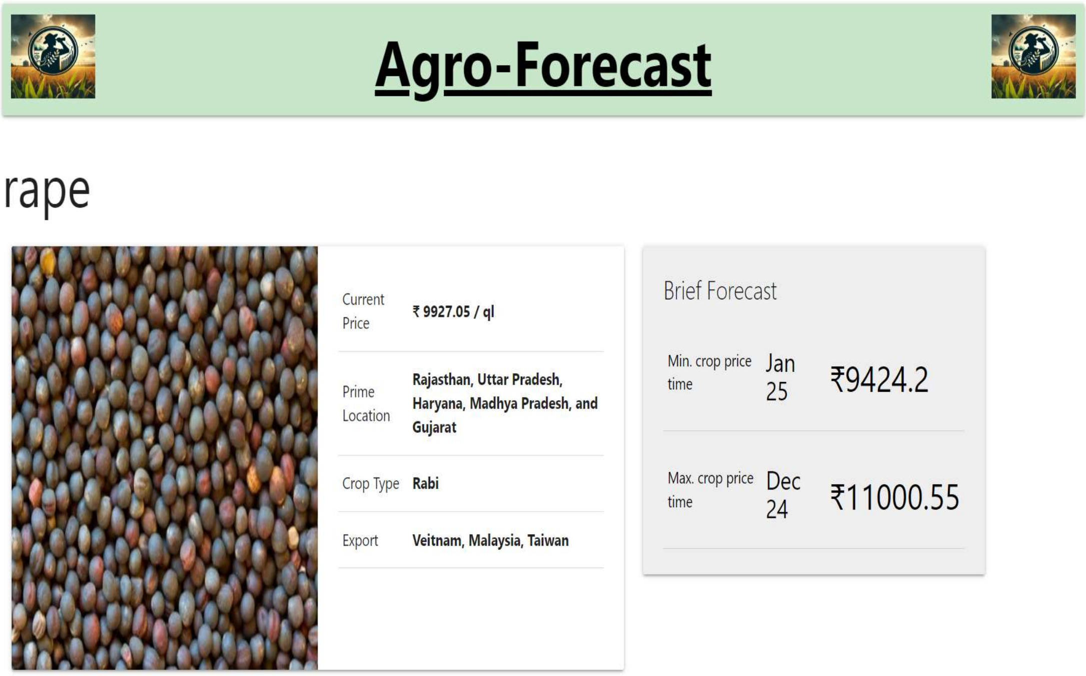


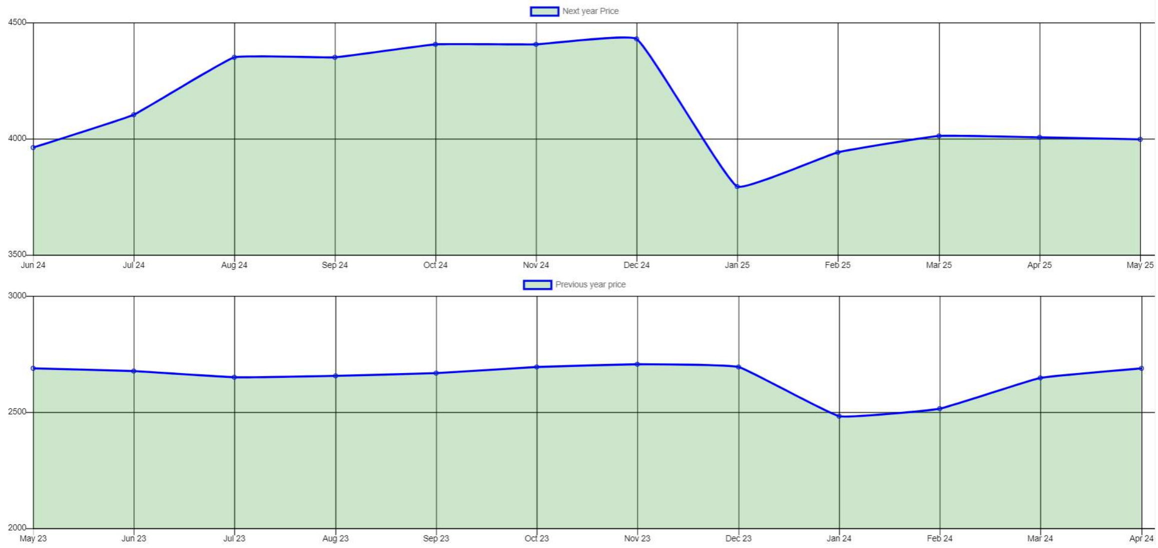
Fig 7.7: BRIEF INFORMATION ON RAPE CROP

Forecast Trends

Month	Price (per Qtl.)	Change
Jun 24	₹9842.3	-0.85% ▼
Jul 24	₹10186.95	2.62% ▲
Aug 24	₹10802.8	8.82% ▲
Sep 24	₹10802.8	8.82% ▲
Oct 24	₹10944.05	10.24% ▲
Nov 24	₹10944.05	10.24% ▲
Dec 24	₹11000.55	10.81% ▲
Jan 25	₹9424.2	-5.07% ▼
Feb 25	₹9785.8	-1.42% ▼
Mar 25	₹9966.6	0.4% ▲
Apr 25	₹9949.65	0.23% ▲
May 25	₹9927.05	0.0% ▲

Fig 7.8: MUSTARD SEED (RAPE CROP'S) 12 MONTHS PREDICTION (WITH GRAPH)

Fig 7.9: BRIEF INFORMATION ON WHEAT CROP



Forecast Trends

Month	Price (per Qtl.)	Change
Jun 24	₹3963.05	-0.85% ▼
Jul 24	₹4101.82	2.62% ▲
Aug 24	₹4349.8	8.82% ▲
Sep 24	₹4349.8	8.82% ▲
Oct 24	₹4406.68	10.24% ▲
Nov 24	₹4406.68	10.24% ▲
Dec 24	₹4429.42	10.81% ▲
Jan 25	₹3794.7	-5.07% ▼
Feb 25	₹3940.3	-1.42% ▼
Mar 25	₹4013.1	0.4% ▲
Apr 25	₹4006.28	0.23% ▲
May 25	₹3997.18	0.0% ▲

Fig 7.10: WHEAT CROP'S 12 MONTHS FORECAST (WITH GRAPH)

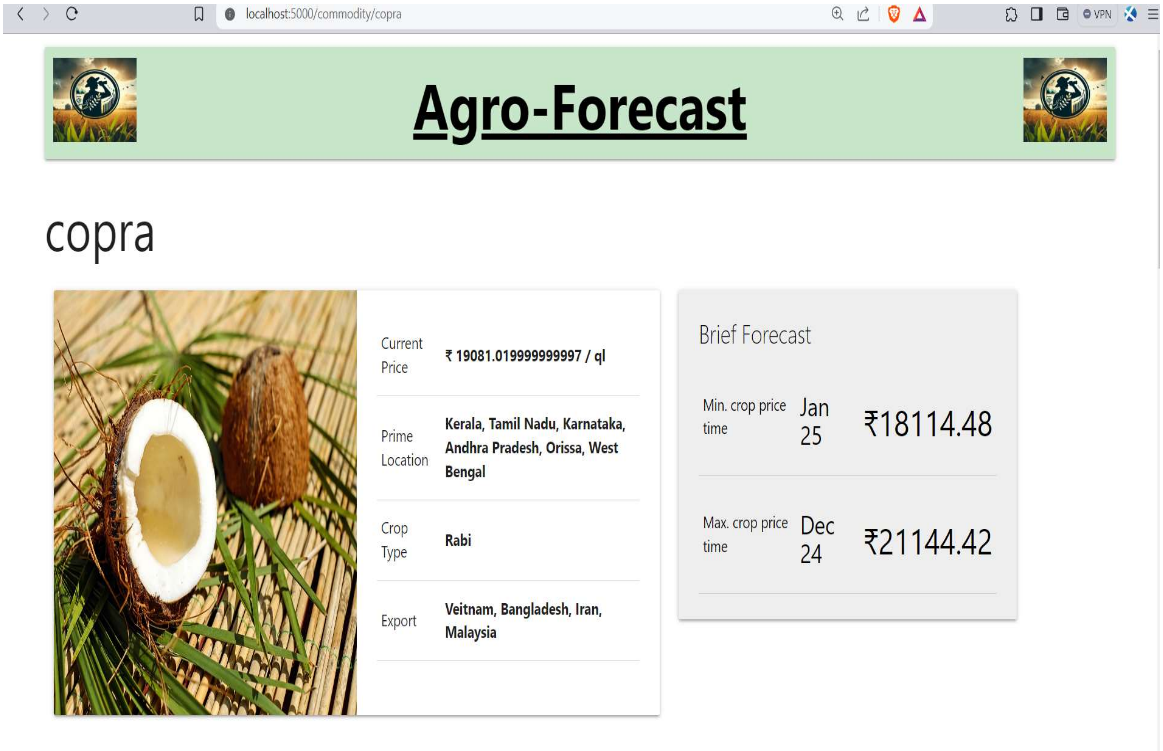
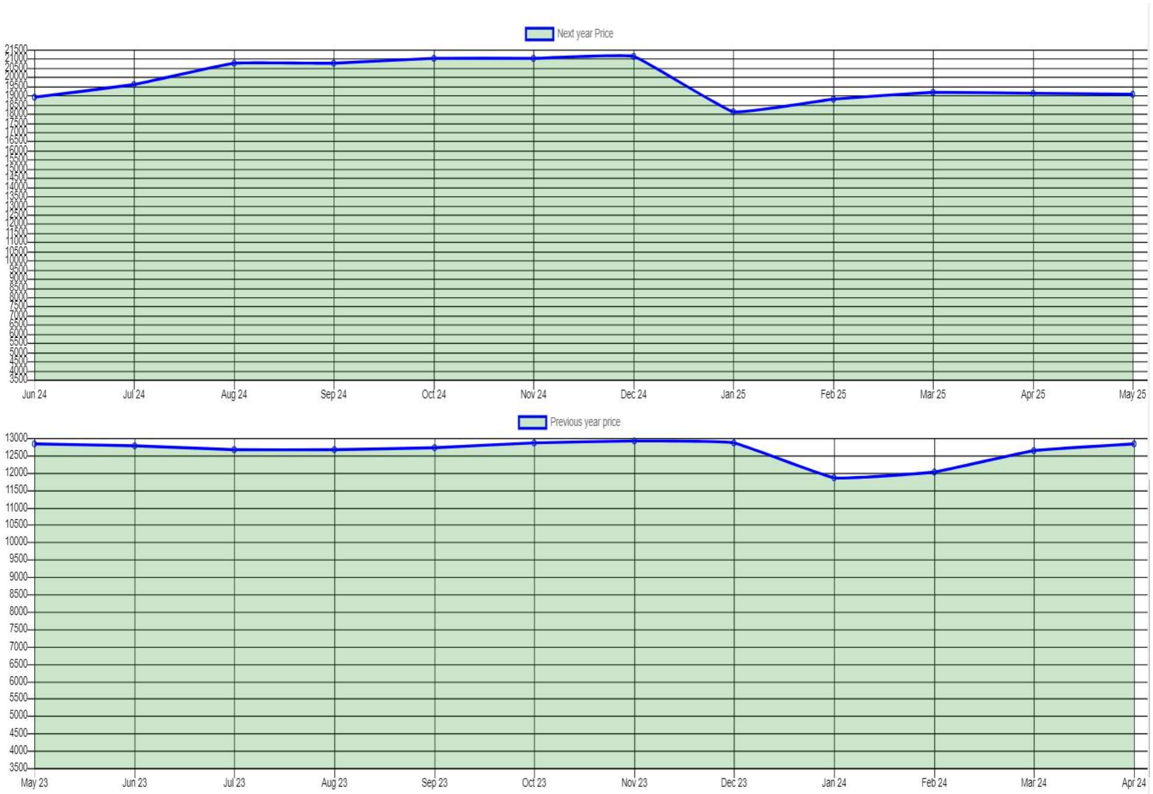


Fig 7.11: BRIEF INFORMATION ON COPRA CROP



Forecast Trends

Month	Price (per Qtl.)	Change
Jun 24	₹18918.12	-0.85% ▼
Jul 24	₹19580.58	2.62% ▲
Aug 24	₹20764.32	8.82% ▲
Sep 24	₹20764.32	8.82% ▲
Oct 24	₹21035.82	10.24% ▲
Nov 24	₹21035.82	10.24% ▲
Dec 24	₹21144.42	10.81% ▲
Jan 25	₹18114.48	-5.07% ▼
Feb 25	₹18809.52	-1.42% ▼
Mar 25	₹19157.04	0.4% ▲
Apr 25	₹19124.46	0.23% ▲
May 25	₹19081.02	0.0% ▲

Fig 7.12: COPRA CROP'S 12 MONTHS FORECAST (WITH GRAPH)

CONCLUSION

The objective of this research is to predict crop prices and profitability before planting using sophisticated machine learning algorithms. The accompanying web application is designed with efficient technologies and a user-friendly interface to facilitate ease of use for farmers. By leveraging comprehensive training datasets, the system can provide valuable insights into market dynamics, enabling accurate predictions of prices and demand.

The primary aim of this initiative is to empower farmers by reducing the uncertainties and challenges they face in agricultural markets. By having access to reliable price and demand forecasts, farmers can make informed decisions regarding their crop selection and planting strategies. This, in turn, can contribute to better financial outcomes and overall stability in the agricultural sector.

Furthermore, the system's potential to mitigate distress among farmers is significant. Agriculture can be a volatile industry, with fluctuations in prices and demand posing serious risks to farmers' livelihoods. By offering tools that enhance predictability and reduce risk, this research contributes to the well-being of farmers and the sustainability of agriculture.

Importantly, the aim to prevent extreme measures such as suicide underscores the critical role of this research. Farming communities often face profound economic pressures, and access to accurate market insights can be a lifeline. By providing farmers with actionable data and reliable forecasts, this initiative seeks to promote resilience and alleviate the desperation that can arise from financial uncertainty in agriculture.

SCOPE FOR FUTURE WORK

By enriching the dataset with additional features, we can enhance the precision of predicting both crop yield and pricing. Incorporating a wider variety of crops into the dataset expands the scope of predictive capabilities, allowing for more comprehensive insights into market trends and agricultural outcomes. The web application's user experience will be optimized with responsive design, engaging animations, and an attractive interface to encourage broader adoption and usability among farmers.

To predict crop prices, various machine learning algorithms like decision trees, support vector machines, neural networks, and deep learning can be employed. Our current model employs a supervised learning algorithm, the Decision Tree Regressor, trained on diverse Kharif and Ragi crops (including Paddy, Wheat, Cotton, Barley, etc.), which enhances accuracy in price forecasting.

In future iterations, the model could integrate climate-aware farming techniques to factor in weather patterns and environmental conditions, offering tailored insights for optimized crop management. Additionally, the model could provide personalized fertilizer recommendations based on soil conditions and crop requirements.

Advanced AI models can be deployed to monitor crop health in real-time, detect pest and disease outbreaks early, and issue proactive alerts to farmers. These innovations are designed to optimize agricultural practices, improve yield forecasts, and empower farmers with actionable insights to make informed decisions.

Ultimately, our goal is to leverage technology to support sustainable and profitable farming practices while mitigating risks and uncertainties faced by farmers. By providing accessible and accurate predictive tools, we aim to enhance productivity, reduce resource waste, and contribute to the overall resilience and prosperity of the agricultural sector.

BIBLIOGRAPHY

https://www.google.com/

https://www.w3schools.com/whatis/whatis_fullstack.asp

Dataset: https://data.gov.in/

Research Paper: https://iarjset.com/

https://www.jetbrains.com/pycharm/

REFERENCES

[1] Ishita Ghutake1,Ritesh Verma1,Rohit Chaudhari1, and Vidhate Amarsinh1,” An Intelligent Crop Price Prediction Using Suitable Machine Learning Algorithm” ITM Web of Conferences 40, 03040 (2021)

[2] Pandit Samuel, B.Sahithi, T.Saheli, D.Ramanika, N.Anil Kumar “Crop Price Prediction System using Machine learning Algorithms." Quest Journals Journal of Software Engineering And Simulation, Vol. 06, No. 01, 2020, Pp. 14-20.

[3] S. Rajeswari, K. Suthendran, “ Developing an Agricultural Product Price Prediction Model using HADT Algorithm”, International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-9 Issue-1S4, December 2019

[4] Gangasagar, Jovin Dsouza, Bhagyashree B Yargal, Arun Kumar, Anuradha Badage, "Crop Price Prediction Using Machine Learning Algorithms ", International Journal of Innovative Research in Science, Engineering and Technology, Volume 9, Issue 10, October 2020.

International Research Journal Of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETS/Certificate/Volume 06/Issue 04/60400272915 Date: 01/05/2024

Certificate of Publication

*This is to certify that author "**Dr. Asha SR**" with paper ID "**IRJMETS60400272915**" has published a paper entitled "CROP PRICE PREDICTION USING MACHINE LEARNING" **in International Research Journal Of Modernization In Engineering Technology And Science (IRJMETS), Volume 06, Issue 04, April 2024***

IRJMETS
Impact Factor
7.868

Editor in Chief

We Wish For Your Better Future

www.irjmets.com

International Research Journal Of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETS/Certificate/Volume 06/Issue 04/60400272915

Date: 01/05/2024

Certificate of Publication

*This is to certify that author "**Saurav Kumar**" with paper ID "**IRJMETS60400272915**" has published a paper entitled "CROP PRICE PREDICTION USING MACHINE LEARNING" **in International Research Journal Of Modernization In Engineering Technology And Science (IRJMETS), Volume 06, Issue 04, April 2024***

A. Deurali

Editor in Chief

IRJMETS
Impact Factor
7.868

We Wish For Your Better Future

www.irjmets.com

IRJMETS

International Research Journal Of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETS/Certificate/Volume 06/Issue 04/60400272915 *Date: 01/05/2024*

Certificate of Publication

*This is to certify that author "**Prafull Kumar Jha**" with paper ID "**IRJMETS60400272915**" has published a paper entitled "CROP PRICE PREDICTION USING MACHINE LEARNING"* ***in International Research Journal Of Modernization In Engineering Technology And Science (IRJMETS), Volume 06, Issue 04, April 2024***

A. Desai

Editor in Chief

IRJMETS
Impact Factor
7.868

International Research Journal of Modernization in Engineering Technology and Science

We Wish For Your Better Future

www.irjmets.com

International Research Journal Of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETS/Certificate/Volume 06/Issue 04/60400272915

Date: 01/05/2024

Certificate of Publication

*This is to certify that author "**Prashant S Patil**" with paper ID "**IRJMETS60400272915**" has published a paper entitled "CROP PRICE PREDICTION USING MACHINE LEARNING"* ***in International Research Journal Of Modernization In Engineering Technology And Science (IRJMETS), Volume 06, Issue 04, April 2024***

IRJMETS Impact Factor 7.868

Editor in Chief

We Wish For Your Better Future

www.irjmets.com

International Research Journal Of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETS/Certificate/Volume 06/Issue 04/60400272915

Date: 01/05/2024

Certificate of Publication

*This is to certify that author "**Shreyas P Bidari**" with paper ID "**IRJMETS60400272915**" has published a paper entitled "CROP PRICE PREDICTION USING MACHINE LEARNING"* ***in International Research Journal Of Modernization In Engineering Technology And Science (IRJMETS), Volume 06, Issue 04, April 2024***

Editor in Chief

IRJMETS Impact Factor 7.868

International Research Journal of Modernization in Engineering Technology and Science

We Wish For Your Better Future

www.irjmets.com